Mobile MultiModal Presentation

Anthony Solon
University of Ulster,
Northland Road, Derry, UK
Tel: +44 2871 371371
aj.solon@ulster.ac.uk

Paul McKevitt
University of Ulster,
Northland Road, Derry, UK
Tel: +44 2871 375433
p.mckevitt@ulster.ac.uk

Kevin Curran
University of Ulster,
Northland Road, Derry, UK
Tel: +44 2871 375565
kj.curran@ulster.ac.uk

ABSTRACT

This paper presents the latest research into a mobile intelligent multimedia presentation system called TeleMorph which can dynamically generate a multimedia presentation using output modalities that are determined by the bandwidth available on a mobile device's wireless connection. To demonstrate the effectiveness of this research TeleTuras, a tourist information guide will implement the solution provided by TeleMorph, thus demonstrating its effectiveness. This paper highlights issues surrounding such a system & introduces the architecture.

Categories and Subject Descriptors

H.5.2 [**Information Interfaces and Presentation**]: User Interfaces - *Haptic I/O, Interaction styles, Natural language*

General Terms

Design, Human Factors

Keywords

Bandwidth determined multimodal presentation, HCI

1 INTRODUCTION

Whereas traditional interfaces support sequential and un-ambiguous input from keyboards and conventional pointing devices, intelligent multimodal interfaces relax these constraints and typically incorporate a broader range of input devices (e.g., spoken language, eye and head tracking, three dimensional (3D) gesture). The integration of multiple modes of input allows users to benefit from the optimal way in which human communication works. Although humans have a natural facility for managing and exploiting multiple input and output media, computers do not. To incorporate multimodality in user interfaces enables computer behaviour to become analogous to human communication paradigms, and therefore the interfaces are easier to learn and use. Since there are large individual differences in ability and preference to use different modes of communication, a multimodal interface permits the user to exercise selection and control over how they interact with the computer.

MM'04, October 10-16, 2004, New York, New York, USA.

In this respect, multimodal interfaces have the potential to accommodate a broader range of users than traditional graphical user interfaces (GUIs) and unimodal interfaces - including users of different ages, skill levels, native language status, cognitive styles, sensory impairments, and other handicaps or illnesses. Interfaces involving spoken or pen-based input, as well as the combination of both, are particularly effective for supporting mobile tasks, such as communications and personal navigation. Unlike the keyboard and mouse, both speech and pen are compact and portable. When combined, people can shift these input modes from moment to moment as environmental conditions change [1]. Implementing multimodal user interfaces on mobile devices is not as clear-cut as doing so on ordinary desktop devices. This is due to the fact that mobile devices are limited in many respects: memory, processing power, input modes, battery power, and an unreliable wireless connection with limited bandwidth. This project will research and implement a framework for Multimodal interaction in mobile environments taking into consideration fluctuating bandwidth. The system output will be bandwidth dependent, with the result that output from semantic representations is dynamically morphed between modalities or combinations of modalities. With the advent of 3G wireless networks and the subsequent increased speed in data transfer available, the possibilities for applications and services that will link people throughout the world who are connected to the network will be unprecedented. One may even anticipate a time when the applications and services available on wireless devices will replace the original versions implemented on ordinary desktop computers. Some projects have already investigated mobile intelligent multimedia systems, using tourism in particular as an application domain. [2] is one such project which analysed and designed a position-aware speech-enabled hand-held tourist information system for Aalborg in Denmark. The main point to note about existing systems is that current mobile intelligent multimedia systems fail to take into consideration network constraints and especially the bandwidth available when transforming semantic representations into the multimodal output presentation. If the bandwidth available to a device is low then it's obviously inefficient to attempt to use video or animations as the output on the mobile device. This would result in an interface with depreciated quality, effectiveness and user acceptance. This is an important issue as regards the usability of the interface. Learnability, throughput, flexibility and user-attitude are the four main concerns affecting the usability of any interface. In the case of the previously mentioned scenario (reduced bandwidth => slower/inefficient output) the throughput of the interface is affected and as a result the user's attitude also. This is only a problem when the required bandwidth for the output modalities exceeds that which is available; hence, the importance of choosing the correct output modality/modalities in relation to available resources.

2 TELEMORPH

The aim of the TeleMorph project is to create a system that dynamically morphs between output modalities depending on available network bandwidth. The aims are to:

- Determine a wireless system's output presentation (unimodal/multimodal) depending on network bandwidth available to device connected to system.
- Implement TeleTuras, a tourist information guide for the city of Derry and integrate the solution provided by TeleMorph, thus demonstrating its effectiveness.

The aims entail the following objectives which include receiving and interpreting questions from the user; Mapping questions to multimodal semantic representation; Matching multimodal representation to database to retrieve answer; Mapping answers to multimodal semantic representation; Querying bandwidth status and generating multimodal presentation based on bandwidth data. The domain chosen as a testbed for TeleMorph is *e*Tourism. It will incorporate route planning, maps, points of interest, spoken presentations, graphics of important objects in the area and animations. The main focus will be on the output modalities used to communicate this information and also the effectiveness of this communication. TeleTuras will be capable of taking input queries in a variety of modalities whether they are combined or used individually. Queries can also be directly related to the user's position and movement direction enabling questions/commands such as:

- "Where is the Leisure Center?"
- "Take me to the Council Offices"
- "What buildings are of interest in this area?"

J2ME (Java 2 Micro Edition) is an ideal programming language for developing TeleMorph, as it is the target platform for the Java Speech API (JSAPI) [3]. The JSAPI enables the inclusion of speech technology in user interfaces for Java applets and applications. The Java Speech API Markup Language [4] and the Java Speech API Grammar Format [4] are companion specifications to the JSAPI. JSML (currently in beta) defines a standard text format for marking up text for input to a speech synthesiser. JSGF version 1.0 defines a standard text format for providing a grammar to a speech recogniser. JSAPI does not provide any speech functionality itself, but through a set of APIs and event interfaces, access to speech functionality provided by supporting speech vendors is accessible to the application. As it is inevitable that a majority of tourists will be foreigners it is necessary that TeleTuras can process multilingual speech recognition and synthesis. To support this an IBM implementation of JSAPI "speech for Java" will be utilised. It supports US&UK English, French, German, Italian, Spanish, and Japanese. To incorporate the navigation aspect of the proposed system a global positioning system connection is incorporated. The User Interface (UI) defined in J2ME is logically composed of two sets of APIs, High-level UI API which emphasises portability across different devices and the Low-level UI API which emphasises flexibility and control. TeleMorph will use a dynamic combination of these in order to provide the best solution possible. An overview of the architecture to date is shown in Figure 1. Media Design takes the output information and morphs it into relevant modality/modalities depending on the information it receives from the Server Intelligent Agent regarding available bandwidth. Media Analysis receives input from the Client device and analyses it to distinguish the modality types that the user utilised in their input. The Domain Model, Discourse Model, User Model, GPS and WWW are additional sources of information for the Multimodal Interaction Manager that assist it in producing an appropriate and correct output presentation.

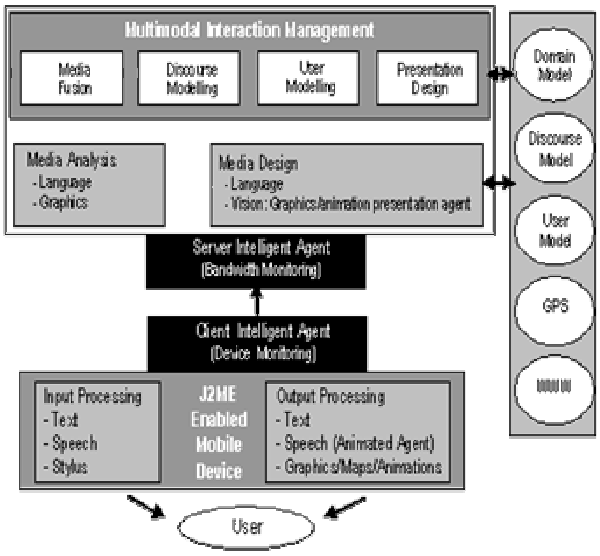

Figure 1: TeleMorph Architecture

The Server Intelligent Agent is responsible for monitoring bandwidth, sending streaming media which is morphed to the appropriate modalities and receiving input from client device & mapping to multimodal interaction manager. The Client Intelligent Agent is in charge of monitoring device constraints e.g. memory, sending multimodal information on input to the server and receiving streamed multimedia.

2.1 Client output

Output on thin client devices connected to TeleMorph will primarily utilise a SMIL media player which will present video, graphics, text and speech to the end user of the system. The J2ME Text-To-Speech (TTS) engine processes speech output to the user. An autonomous agent will be integrated into the TeleMorph client for output as they serve as an invaluable interface agent to the user as they incorporate modalities that are the natural modalities of face-to-face communication among humans. A SMIL media player will output audio on the client device. This audio will consist of audio files that are streamed to the client when the necessary bandwidth is available. However, when sufficient bandwidth is unavailable audio files will be replaced by ordinary text which will be processed by a TTS engine on the client producing synthetic speech output.

2.2 Autonomous agents

An autonomous agent will serve as an interface agent to the user as they incorporate modalities that are the natural modalities of face-to-face communication among humans. It will assist in communicating information on a navigation aid for tourists about sites, points of interest, and route planning. Microsoft Agent[1] provides a set of programmable software services that supports the presentation of interactive animated characters. It enables developers to incorporate conversational interfaces, which leverage natural aspects of human social communication. In addition to mouse and keyboard input, Microsoft Agent includes support for speech recognition so applications can respond to voice commands. Characters can respond using synthesised speech, recorded audio, or text. One advantage of agent characters is they provide higher-levels of a character's movements often found in the performance arts, like blink, look up, look down, and walk. BEAT, another animator's tool which was incorporated in REA, allows animators to input typed text that they wish to be spoken by an animated figure.

2.3 Client input

The TeleMorph client will allow for speech recognition, text and haptic deixis (touch screen) input. A speech recognition engine will be reused to process speech input from the user. Text and haptic input will be processed by the J2ME graphics API. Speech recognition in TeleMorph resides in *Capture Input* as illustrated in figure 2. The Java Speech API Mark-up Language[2] defines a standard text format for marking up text for input to a speech synthesiser. As mentioned before JSAPI does not provide any speech functionality itself, but through a set of APIs and event interfaces, access to speech functionality (provided by supporting speech vendors) is accessible to the application.

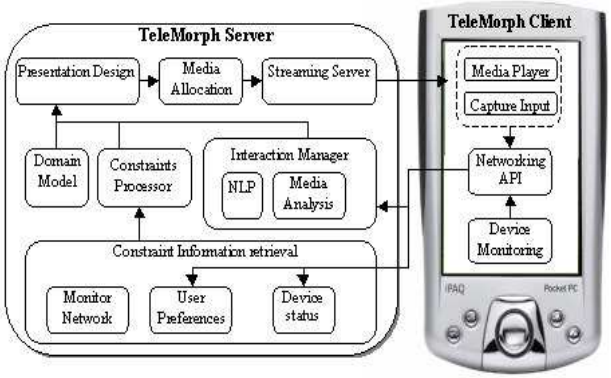

Figure 2: Modules within TeleMorph

For this purpose IBM's implementation of JSAPI "speech for Java" is adopted for providing multilingual speech recognition functionality. This implementation of the JSAPI is based on ViaVoice, which will be positioned remotely in the *Interaction Manager* module on the server. The relationship between the JSAPI speech recogniser (in the *Capture Input* module on the client and ViaVoice (in the *Interaction Manager* on the server is necessary as speech recognition is computationally too heavy to be processed on a thin client. After the ViaVoice speech recogniser has processed speech which is input to the client device, it will also need to be analysed by an *NLP* module to assess its semantic content. A reusable tool to do this is yet to be decided upon to complete this task. Possible solutions for this include adding an additional NLP component to ViaVoice; or perhaps reusing other natural understanding tools such as PC-PATR [5] which is a natural language parser based on context-free phrase structure grammar and unifications on the feature structures associated with the constituents of the phrase structure rules.

2.4 Graphics

The User Interface (UI) defined in J2ME is logically composed of two sets of APIs, High-level UI API which emphasises portability across different devices and the Low-level UI API which emphasises flexibility and control. The portability in the high-level API is achieved by employing a high level of abstraction. The actual drawing and processing user interactions are performed by implementations. Applications that use the high-level API have little control over the visual appearance of components, and can only access high-level UI events. On the other hand, using the low-level API, an application has full control of appearance, and can directly access input devices and handle primitive events generated by user interaction. However the low-level API may be device-dependent, so applications developed using it will not be portable to other devices with a varying screen size. TeleMorph uses a combination of these to provide the best solution possible. Using these graphics APIs, TeleMorph implements a *Capture Input* module which accepts text from the user. Also using these APIs, haptic input is processed by the *Capture Input* module to keep track of the user's input via a touch screen, if one is present on the device. User preferences in relation to modalities and cost incurred are managed by the *Capture Input* module in the form of standard check boxes and text boxes available in the J2ME high level graphics API.

2.5 Networking

Networking takes place using sockets in the *J2ME Networking API* module as shown in figure 2 to communicate data from the *Capture Input* module to the *Media Analysis* and *Constraint Information Retrieval* modules on the server. Information on client device constraints will also be received from the *Device Monitoring* module to the *Networking API* and sent to the relevant modules within the *Constraint Information Retrieval* module on the server. Networking in J2ME has to be very flexible to support a variety of wireless devices and has to be device specific at the same time. To meet this challenge, the Generic Connection Framework (GCF) is incorporated into J2ME. The idea of the GCF is to define the abstractions of the networking and file input/output as generally as possible to support a broad range of devices, and leave the actual implementations of these abstractions to the individual device manufacturers. These abstractions are defined as Java interfaces. The device manufacturers choose which one to implement based on the actual device capabilities.

[1] http://www.microsoft.com/msagent/default.asp
[2] http://java.sun.com/products/java-media/speech/

2.6 TeleMorph Server-Side

SMIL is utilised to form the semantic representation language in TeleMorph and will be processed by the *Presentation Design* module in figure 2. The HUGIN development environment [6] allows TeleMorph to develop its decision making process using Causal Probabilistic Networks which will form the *Constraint Processor* module. The ViaVoice speech recognition software resides within the *Interaction Manager* module. On the server end of the system Darwin streaming server[3] is responsible for transmitting the output presentation from the TeleMorph server application to the client's *Media Player*.

2.6.1 SMIL semantic representation

The XML based SMIL language forms the semantic representation language of TeleMorph used in the *Presentation Design* module as shown in figure 2. TeleMorph designs SMIL content that comprises multiple modalities that exploit currently available resources fully, whilst considering various constraints that affect the presentation, but in particular, bandwidth. This output presentation is then streamed to the *Media Player* module on the mobile client for displaying to the end user. TeleMorph will constantly recycle the presentation SMIL code to adapt to continuous and unpredictable variations of physical system constraints (e.g. fluctuating bandwidth, device memory), user constraints (e.g. environment) and user choices (e.g. streaming text instead of synthesised speech). In order to present the content to the user, a SMIL media player needs to be available on the client device.

2.6.2 TeleMorph reasoning - CPNs/BBNs

Causal Probabilistic Networks aid in conducting reasoning and decision making within the *Constraints Processor* module. In order to implement Bayesian Networks in TeleMorph, the HUGIN [6] development environment is used. HUGIN provides the necessary tools to construct Bayesian Networks. When a network has been constructed, one can use it for entering evidence in some of the nodes where the state is known and then retrieve the new probabilities calculated in other nodes corresponding to this evidence. A Causal Probabilistic Network (CPN)/Bayesian Belief network (BBN) is used to model a domain containing uncertainty in some manner. It consists of a set of nodes and a set of directed edges between these nodes. A Belief Network is a Directed Acyclic Graph (DAG) where each node represents a random variable. Each node contains the states of the random variable it represents and a conditional probability table (CPT) or, in more general terms, a conditional probability function (CPF). The CPT of a node contains probabilities of the node being in a specific state given the states of its parents. Edges reflect cause-effect relations within the domain. These effects are normally not completely deterministic (e.g. disease -> symptom). The strength of an effect is modelled as a probability.

2.6.3 JATLite middleware

As TeleMorph is composed of several modules with different tasks to accomplish, the integration of the selected tools to complete each task is important. To allow for this a middleware is required within the *TeleMorph Server*. One such middleware is JATLite (Jeon et al. 2000) which was developed by the Stanford University. JATLite provides a set of Java packages which makes it easy to build multi-agent systems using Java. Different layers are incorporated to achieve this, including:

- Abstract layer- provides a collection of abstract classes necessary for JATLite implementation. Although JATLite assumes all connections to be made with TCP/IP, the abstract layer can be extended to implement different protocols such as UDP.

- Base layer- provides communication based on TCP/IP and the abstract layer. There is no restriction on the message language or protocol. The base layer can be extended, for example, to allow inputs from sockets and output to files. It can also be extended to give agents multiple message ports.

- KQML (Knowledge Query & Manipulation Language) layer provides for storage & parsing of KQML messages

[3] http://developer.apple.com/darwin/projects/darwin/

3 CONCLUSION

This paper has presented a Mobile Intelligent System called TeleMorph that dynamically morphs between output modalities depending on available network bandwidth. TeleMorph will be able to dynamically generate a multimedia presentation from semantic representations using output modalities that are determined by constraints that exist on a mobile device's wireless connection, the mobile device itself and also those limitations experienced by the end user of the device. The output presentation will include Language and Vision modalities consisting of video, speech, non-speech audio and text. Input to the system will be in the form of speech, text and haptic deixis.

4 REFERENCES

[1] Holzman, T (1999) Computer-human interface solutions for emergency medical care. *Interactions, 6(3),* 13-24.

[2] Koch, U.O. (2000) Position-aware Speech-enabled Hand Held Tourist Information System. Internal report, Aalborg University, Institute of Electronic Systems, Denmark.

[3] JCP (2002) Java Community Process. http://www.jcp.org/

[4] JSML & JSGF (2002). Java Community Process. http://www.jcp.org/en/home/index Site visited 30/09/2003

[5] McConnel, S. (1996) KTEXT and PC-PATR: Unification based tools for computer aided adaptation. In *Proceedings of the 1996 general CARLA conference*, November 14-15, 39-95. Dallas: JAARS and Summer Institute of Linguistics.

[6] HUGIN (2003) http://www.hugin.com/

[7] Jeon, H., C. Petrie & M.R. Cutkosky (2000) JATLite: A Java Agent Infrastructure with Message Routing. IEEE Internet Computing Vol. 4, No. 2, Mar/Apr, 87-96.